

# Covariate Selection in High-Dimensional Generalized Linear Models With Measurement Error

Øystein Sørensen<sup>\*1</sup>, Arnoldo Frigessi<sup>†1,2</sup>, and Magne Thoresen<sup>‡1</sup>

<sup>1</sup>Oslo Centre for Biostatistics and Epidemiology, Department of Biostatistics, University of Oslo, Norway

<sup>2</sup>Oslo Centre for Biostatistics and Epidemiology, Research Support Services, Oslo University Hospital, Norway

July 7, 2014

## Abstract

In many problems involving generalized linear models, the covariates are subject to measurement error. When the number of covariates  $p$  exceeds the sample size  $n$ , regularized methods like the lasso or Dantzig selector are required. Several recent papers have studied methods which correct for measurement error in the lasso or Dantzig selector for linear models in the  $p > n$  setting. We study a correction for generalized linear models based on Rosenbaum and Tsybakov's matrix uncertainty selector. By not requiring an estimate of the measurement error covariance matrix, this generalized matrix uncertainty selector has a great practical advantage in problems involving high-dimensional data. We further derive an alternative method based on the lasso, and develop efficient algorithms for both methods. In our simulation studies of logistic and Poisson regression with measurement error, the proposed methods outperform the standard lasso and Dantzig selector with respect to covariate selection, by reducing the number of false positives considerably. We also consider classification of patients on the basis of gene expression data with noisy measurements.

## 1 Introduction

Regularization methods like the lasso (Tibshirani, 1996) and Dantzig selector (DS) (Candes and Tao, 2007) are widely used in regression, when the number

---

<sup>\*</sup>oystein.sorensen@medisin.uio.no

<sup>†</sup>arnoldo.frigessi@medisin.uio.no

<sup>‡</sup>magne.thoresen@medisin.uio.no

of covariates  $p$  exceeds the number of measurements  $n$ . Lasso and DS typically select a small set of covariates which are given nonzero estimated regression coefficients. In addition to yielding models with good predictive performance, the set of selected covariates can be a good starting point for further scientific investigations. For example, in classification of cancer patients based on gene expression measurements, the selected genes can be interesting candidates for further study. The covariate selection and parameter estimation properties of the lasso and the closely related DS have received much attention in the literature, e.g., Bickel et al. (2009); Meinshausen and Bühlmann (2006); van de Geer and Bühlmann (2009); Wainwright (2009); Zhao and Yu (2006), summarized in the recent monograph Bühlmann and van de Geer (2011). Bounding the estimation error of the regression coefficients with high probability requires that the covariates satisfy a restricted eigenvalue condition (Bickel et al., 2009), while perfectly recovering the set of nonzero coefficients with the lasso requires a stronger irrepresentable condition (Meinshausen and Bühlmann, 2006; Zou, 2006; Zhao and Yu, 2006).

Both the lasso and the DS can be defined for generalized linear models (GLMs) (McCullagh and Nelder, 1989). The framework of GLMs includes logistic regression for binomial outcomes and Poisson regression for outcomes which can be modeled by a Poisson process. The extension of lasso to GLMs was derived in the original lasso paper (Tibshirani, 1996), while the generalized DS (GDS) has been proposed by James and Radchenko (2009). For the lasso, the theoretical results for linear regression models have been extended to GLMs (Bühlmann and van de Geer, 2011; van de Geer, 2008).

An implicit assumption in all the work mentioned so far, is that the covariates are perfectly measured, i.e., that the only source of noise is the stochastic relationship between the measured covariates and the outcome. In reality, most problems are subject to at least a small amount of measurement error. For example, gene expression microarray measurements are subject to various sources of systematic and random error (Rocke and Durbin, 2001), and are a noisy version of the true gene expression in the patients. Another example is food frequency questionnaires (FFQs) used in epidemiologic studies, in which subjects are asked about their food consumption. It is well known that the responses given in FFQs are often far away from the true food consumption (Kipnis et al., 2003). Sensor network data also tend to be noisy, due to measurement error or sensor failure (Bertrand and Moonen, 2011).

In classical regression problem ( $p < n$ ), measurement error is known to yield biased parameter estimates and lack of power (Carroll et al., 2006). Correction methods typically require replicate measurements of the covariates in order to estimate the measurement error distribution. We refer to the excellent monograph Carroll et al. (2006) for an overview of measurement error models in statistics. Correction for measurement error in penalized regression has been studied by various authors recently (Chen and Caramanis, 2013; Liang and Li, 2009; Loh and Wainwright, 2012; Ma and Li, 2010; Rosenbaum and Tsybakov, 2010, 2013; Sørensen et al., 2014; Zhu et al., 2011). Other authors have considered measurement error in the response rather than in the covariates, e.g.,

Nguyen and Tran (2013). Under a sparsity assumption, it has been shown that the standard formulations of the lasso and DS may select too many covariates in the presence of measurement error (Rosenbaum and Tsybakov, 2010; Sørensen et al., 2014). The correction methods studied by Liang and Li (2009); Loh and Wainwright (2012); Ma and Li (2010); Rosenbaum and Tsybakov (2013); Sørensen et al. (2014) require knowledge of the measurement error distribution. These likelihood based approaches yield estimators with good statistical properties. However, in many applications with large  $p$ , it may be hard even to obtain a rough estimate of the  $p \times p$  covariance matrix of the measurement error, which would be needed in the case of Gaussian additive errors. In contrast, the matrix uncertainty selector (MUS) of Rosenbaum and Tsybakov (2010), the sparse total least squares (S-TLS) method of Zhu et al. (2011), and the orthogonal matching pursuit (OMP) algorithm of Chen and Caramanis (2013), account for the measurement error without requiring an estimate of its distribution. The latter methods will hence have a practical advantage in many applications, and in particular, they have been shown to yield fewer false positive (FP) selections than the standard lasso and DS.

The methods proposed for dealing with measurement error in penalized regression all focus on linear regression, with the exception of Ma and Li (2010) and Sørensen et al. (2014). Considering the importance and general applicability of GLMs, it is therefore of interest to develop penalized regression methods for GLMs which do not require an estimate of the measurement error distribution, and recover the linear regression framework as a special case. In this paper, we propose the generalized MUS (GMUS), based on a Taylor expansion of the GLM mean function around the true, but unknown, covariates. The GMUS can be computed using iteratively reweighted least squares (IRLS), and when the Taylor expansion is truncated at first order, each step of the IRLS algorithm requires solving a simple linear program. We also develop a generalized matrix uncertainty lasso (GMUL), which is a lasso-type analog of the DS-based GMUS. The GMUL estimate can also be computed using an IRLS algorithm, in which an inner coordinate descent loop has to be run until convergence in each step of the algorithm. In simulation experiments with logistic and Poisson regression, the GMUL and GMUS with the first order Taylor approximation are shown to give very promising covariate selection results compared to the lasso and the GDS, by detecting considerable fewer false positives (FPs) at similar numbers of true positives (TPs).

The outline of the paper is as follows. In Section 2 we define the setup and notation, and give a brief overview of the relevant background. In Section 3 we derive the GMUS for GLMs with measurement error as an extension of the MUS for linear models. We also describe an IRLS algorithm for computing the GMUS. Next, in Section 4 we show how a lasso analog of the GMUS can be defined through the KKT conditions, which we term the GMUL. An algorithm for computing the GMUL is developed, which requires an outer IRLS loop and an inner coordinate descent loop. In Section 5 we investigate the convergence of our algorithms for the GMUL and the GMUS in a simple numerical experiment. In Section 6 we present the results of simulation experiments comparing both

methods to the standard formulations of the GDS and the lasso for logistic and Poisson regression. The simulation studies confirm earlier results showing that the standard methods grossly overestimate the number of nonzero regression coefficients in the presence of measurement error. In the settings considered, the GMUL and GMUS reduce the number of false positive selections considerably, without missing many true positives. Finally, in Section 7 we apply the GMUL to a problem of finding genes which are differentially expressed between patients with low vs. high bone mineral density. Mathematical derivations are given in the appendices.

## 2 Background and Model Setup

We consider a GLM (McCullagh and Nelder, 1989) with response  $Y$  distributed according to

$$f_Y(y; \theta, \phi) = \exp \left\{ \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right\}, \quad (1)$$

with linear predictor  $\theta = \mathbf{x}^T \boldsymbol{\beta}^0$ . The covariates are  $\mathbf{x} \in \mathbb{R}^p$ , the set of nonzero coefficients  $S^0 = \{j : \beta_j^0 \neq 0\}$  has cardinality  $s = \text{card}\{S^0\}$ , and  $\boldsymbol{\beta}^0 \in \mathbb{R}^p$  is the true vector of regression coefficients. In order to accomodate an intercept term, we implicitly assume that the first element of  $\mathbf{x}$  is a constant 1. The expected response is given by the mean function

$$\mu(\theta) = b'(\theta) = g^{-1}(\theta),$$

where  $g^{-1}(\cdot)$  is the inverse of a canonical link function  $g(\cdot)$ . For logistic regression, the mean function equals  $(1 + \exp(-\theta))^{-1}$ , and for Poisson regression it equals  $\exp(\theta)$ . We focus on cases in which the dispersion parameter  $\phi$  is known or a constant, as it is for Poisson and logistic regression, and will hence neglect  $a(\phi)$  for notational convenience in the sequel.

Given a data set with covariates  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$ , and response  $\mathbf{y} = (y_1, \dots, y_n)^T$  distributed according to (1), the lasso (Tibshirani, 1996) estimates  $\boldsymbol{\beta}^0$  by maximizing the log-likelihood subject to an  $\ell_1$ -constraint on the regression coefficients. This yields a nonsmooth convex optimization problem, whose Lagrangian version is

$$\hat{\boldsymbol{\beta}}_L \in \arg \min_{\boldsymbol{\beta}} \frac{-1}{n} \sum_{i=1}^n \{y_i \mathbf{x}_i^T \boldsymbol{\beta} - b(\mathbf{x}_i^T \boldsymbol{\beta})\} + \lambda \|\boldsymbol{\beta}\|_1 \quad (2)$$

where  $\lambda$  is a regularization parameter. An alternative estimator is the GDS,

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{DS} \in \arg \min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1, \text{ subject to (s.t.)} \\ \frac{1}{n} \max_{1 \leq j \leq p} \left| \sum_{i=1}^n x_{ij} \{y_i - \mu(\mathbf{x}_i^T \boldsymbol{\beta})\} \right| \leq \lambda \end{aligned} \quad (3)$$

which can be computed by iterative linear programming using an iteratively reweighted least squares (IRLS) approach (James and Radchenko, 2009). The GDS and the lasso are closely related. For a given  $\lambda$ , we have

$$\hat{\beta}_{DS}, \hat{\beta}_L \in \left\{ \beta \in \mathbb{R}^p : \frac{1}{n} \max_{1 \leq j \leq p} \left| \sum_{i=1}^n x_{ij} \{y_i - \mu(\mathbf{x}_i^T \beta)\} \right| \leq \lambda \right\},$$

and hence, both for the lasso and for the GDS, the correlation of any covariate with the residual is bounded by  $\lambda$ . By definition,  $\|\hat{\beta}_{DS}\|_1 \leq \|\hat{\beta}_L\|_1$  (Bickel et al., 2009).

## 2.1 Covariate Measurement Error

In (2) and (3), it is implicitly assumed that the covariates are perfectly known. In many real applications, however, the measurement process is subject to noise and covariates may be missing at random. In particular, we consider the additive measurement error model

$$\mathbf{w}_i = \mathbf{x}_i + \mathbf{u}_i, \quad i = 1, \dots, n, \quad (4)$$

where  $\mathbf{w}_i$  is the vector of measurements of sample  $i$  and  $\mathbf{u}_i$  is the vector of measurement errors. Problems with missing data can also be expressed in this additive form (Rosenbaum and Tsybakov, 2010). We further assume that the measurements in  $\mathbf{w}_i$  are standardized to have zero mean and unit variance,

$$\frac{1}{n} \sum_{i=1}^n w_{ij} = 0 \text{ and } \frac{1}{n} \sum_{i=1}^n w_{ij}^2 = 1, \quad j = 1, \dots, p, \quad (5)$$

and let  $\mathbf{W} = (w_1, \dots, w_n)^T$  and  $\mathbf{U} = (u_1, \dots, u_n)^T$ . Finally let

$$\epsilon_i = y_i - \mu(\mathbf{x}_i^T \beta^0), \quad i = 1, \dots, n$$

denote the generalized residual and define the vector  $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)^T$ .

In the absence of measurement error,  $\beta^0$  is contained in the feasible set of the GDS if  $\lambda$  is chosen such that

$$\frac{1}{n} \|\mathbf{X}^T \boldsymbol{\epsilon}\|_\infty \leq \lambda \quad (6)$$

holds (Antoniadis et al., 2010; Candes and Tao, 2007). Hence, for any GDS solution  $\hat{\beta}_{DS}$  as well as for the true regression coefficients  $\beta^0$ , the maximum correlation of any covariate with the residual is bounded by  $\lambda$ . Under restricted eigenvalue conditions, this also means that  $\hat{\beta}_{DS}$  and  $\beta^0$  are close, and that their maximum possible distance increases in  $\lambda$ . Hence, a natural starting point for a theoretical analysis is to assume that the bound (6) holds (Bühlmann and van de Geer, 2011, p. 103). However, when the true covariates are unknown, and noisy measurement  $\mathbf{W}$  are plugged into the GDS,  $\beta^0$  is not guaranteed to

be feasible even when (6) holds. The reason is that (6) only bounds the noise corresponding to the residual, whereas the noise in the measured covariates is not taken into account. For the special case of linear regression, Rosenbaum and Tsybakov (2010) introduced a new parameter,  $\delta$ , which bounds the magnitude of the measurement error, yielding the two bounds

$$\frac{1}{n} \|\mathbf{W}^T \boldsymbol{\epsilon}\|_\infty \leq \lambda \text{ and } \|\mathbf{U}\|_\infty \leq \delta, \quad (7)$$

where  $\|\cdot\|_\infty$  is the maximum component norm. When the bounds (7) hold,  $\boldsymbol{\beta}^0$  is a feasible solution of the MUS, given by

$$\hat{\boldsymbol{\beta}}_{MU} \in \arg \min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1, \text{ s.t.} \quad (8)$$

$$\frac{1}{n} \|\mathbf{W}^T (\mathbf{y} - \mathbf{W}\boldsymbol{\beta})\|_\infty \leq \lambda + \delta \|\boldsymbol{\beta}\|_1.$$

The MUS thus modifies the DS by adding the term  $\delta \|\boldsymbol{\beta}\|_1$  to the upper bound on the correlation of the measurements with the residual, ensuring that  $\boldsymbol{\beta}^0$  is still feasible. When  $\delta$  is small, the MUS yields tight bounds for  $\|\hat{\boldsymbol{\beta}}_{MU} - \boldsymbol{\beta}^0\|_q$ ,  $q \in \{1, 2\}$ , as well as good covariate selection properties (Rosenbaum and Tsybakov, 2010). We finally note that the price to pay for not knowing the measurement error distribution, is that the estimation error bounds of the MUS do not go to zero when  $n \rightarrow \infty$ . However, if the main goal is covariate selection, the MUS may still have very good finite sample performance, as illustrated by Rosenbaum and Tsybakov (2010).

### 3 MUS for High-Dimensional GLMs with Measurement Error

We now present an extension of the MUS to GLMs with additive measurement error. In order to do this, we need to find a feasible set which contains  $\boldsymbol{\beta}^0$  and yields good model fit, and then estimate  $\boldsymbol{\beta}^0$  by a sparse element of this set.

First, consider a Taylor expansion of the mean function  $\mu(\theta_i) = \mu(\mathbf{x}_i^T \boldsymbol{\beta}^0)$  around the scalar point  $\mathbf{w}_i^T \boldsymbol{\beta}^0$ ,

$$\mu(\mathbf{x}_i^T \boldsymbol{\beta}^0) = \sum_{r=0}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r, \quad (9)$$

where  $\mu^{(r)}(\cdot)$  denotes the  $r$ th derivate of  $\mu(\cdot)$ . We now have the following result.

**Proposition 1** *Assuming the bounds (7) hold, the true regression coefficients satisfy  $\boldsymbol{\beta}^0 \in \Theta$ , where*

$$\Theta = \left[ \boldsymbol{\beta} \in \mathbb{R}^p : \frac{1}{n} \max_{1 \leq j \leq p} \left| \sum_{i=1}^n w_{ij} \{y_i - \mu(\mathbf{w}_i^T \boldsymbol{\beta})\} \right| \right]$$

$$\leq \lambda + \sum_{r=1}^{\infty} \frac{\delta^r}{r! \sqrt{n}} \|\beta\|_1^r \left\| \mu^{(r)}(\mathbf{W}\beta) \right\|_2 \Bigg],$$

and

$$\mu^{(r)}(\mathbf{W}\beta) = \left( \mu^{(r)}(\mathbf{w}_1^T \beta), \dots, \mu^{(r)}(\mathbf{w}_n^T \beta) \right)^T.$$

The proof is given in Appendix A.

The term

$$\sum_{i=1}^n w_{ij} \{y_i - \mu(\mathbf{w}_i^T \beta)\}$$

in the definition of  $\Theta$  above, is the score function of the log-likelihood with respect to  $\beta$ . In the classical ( $p < n$ ) case, without measurement error, one would typically set  $\lambda = \delta = 0$ , yielding the maximum likelihood estimate. Since  $p > n$ , and the maximum likelihood estimate is not well defined, we allow  $\lambda > 0$ , to an extent which depends on the magnitude of the residuals  $\epsilon$ . This means that the score differs from 0 by at most  $\lambda$ . Furthermore, since we have covariate measurement error, we allow the score to be further away from zero, at a magnitude which depends on the complicated sum involving the regularization parameter  $\delta$ . Finally, we obtain a sparse estimate of the regression function by picking a sparse element of  $\Theta$  according to some criterion. In analog with the GDS, we define the GMUS estimate as the element in  $\Theta$  of minimum L1-norm. Using a finite number  $R$  of terms in the Taylor expansion (9), this yields the following definition of the GMUS:

$$\hat{\beta}_{MU}^R \in \arg \min_{\beta} \|\beta\|_1, \text{ s.t. } \beta \in \Theta^R, \text{ where} \quad (10)$$

$$\Theta^R = \left[ \beta \in \mathbb{R}^p : \frac{1}{n} \max_{1 \leq j \leq p} \left| \sum_{i=1}^n w_{ij} \{y_i - \mu(\mathbf{w}_i^T \beta)\} \right| \right. \\ \left. \leq \lambda + \sum_{r=1}^R \frac{\delta^r}{r! \sqrt{n}} \|\beta\|_1^r \left\| \mu^{(r)}(\mathbf{W}\beta) \right\|_2 \right]. \quad (11)$$

In the rest of this paper, we use  $R = 1$ , and we will show that this first order approximation yields good results in practice.

### 3.1 Computation Using IRLS

We suggest using a weighted least squares approach to compute  $\hat{\beta}_{MU}^R$ . Extending the IRLS algorithm for the GDS (James and Radchenko, 2009), we assume that  $\mathbf{w}_i^T \beta^{(k)}$  is the current estimate of the linear predictor for sample  $i$  after completing the  $k$ th iteration. Now define the adjusted dependent covariate

$$z_i = \mathbf{w}_i^T \beta^{(k)} + \frac{y_i - \mu(\mathbf{w}_i^T \beta^{(k)})}{\mu'(\mathbf{w}_i^T \beta^{(k)})}, \quad i = 1, \dots, n. \quad (12)$$

Next, we define a weight vector for each term in the Taylor expansion with  $R$  terms,

$$\mathbf{V}^{(r)} = \left\{ \mu^{(r)} \left( \mathbf{w}_1^T \boldsymbol{\beta}^{(k)} \right), \dots, \mu^{(r)} \left( \mathbf{w}_n^T \boldsymbol{\beta}^{(k)} \right) \right\}^T, \quad (13)$$

for  $r = 1, \dots, R$ , and introduce the matrix  $\tilde{\mathbf{W}} \in \mathbb{R}^{n \times p}$  and the vector  $\tilde{\mathbf{z}} \in \mathbb{R}^n$  with elements

$$\tilde{w}_{ij} = \sqrt{V_i^{(1)}} w_{ij} \text{ and } \tilde{z}_i = \sqrt{V_i^{(1)}} z_i, \quad (14)$$

for  $i = 1, \dots, n$ ,  $j = 1, \dots, p$ . The next iterate  $\boldsymbol{\beta}^{(k+1)}$  is now given by

$$\boldsymbol{\beta}^{(k+1)} \in \arg \min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1, \text{ s.t.} \quad (15)$$

$$\frac{1}{n} \left\| \tilde{\mathbf{W}}^T (\tilde{\mathbf{z}} - \tilde{\mathbf{W}} \boldsymbol{\beta}) \right\|_{\infty} \leq \lambda + \sum_{r=1}^R \frac{\delta^r}{r! \sqrt{n}} \|\boldsymbol{\beta}\|_1^r \left\| \mathbf{V}^{(r)} \right\|_2$$

When we use a first order Taylor expansion ( $R = 1$ ), (15) is equivalent to a linear program, as shown in Appendix B. The procedure is summarized in the following algorithm. As initial estimate  $\boldsymbol{\beta}^{(0)}$ , a convenient choice may be to take the GDS solution (3), corresponding to  $\delta = 0$ .

**Require:** Fix an initial estimate  $\boldsymbol{\beta}^{(0)}$ ,  $k = 0$ ,  $R \in \mathbb{N}$ .

- 1: **repeat**
- 2:   Compute  $\mathbf{V}^{(r)}$  according to (13)
- 3:   Compute  $\mathbf{z}$  according to (12)
- 4:   Compute  $\tilde{\mathbf{W}}$  and  $\tilde{\mathbf{z}}$  according to (14)
- 5:   Compute  $\boldsymbol{\beta}^{(k+1)}$  by solving (15)
- 6:    $k \leftarrow k + 1$
- 7: **until**  $\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(k-1)}\| < \epsilon_{tol}$
- 8: **return**  $\hat{\boldsymbol{\beta}}_{MU}^R = \boldsymbol{\beta}^{(k)}$

On convergence,  $\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)}$  up to a small tolerance parameter  $\epsilon_{tol}$ , and  $\boldsymbol{\beta}^{(k+1)}$  is solution of the GMUS (10).

This algorithm is not guaranteed to converge, because the GMUS will generally have multiple solutions. In our simulation experiments with  $n = 200$  and  $p = 500$ , we have experienced that it converges in a small number of iterations except when  $\lambda$  and  $\delta$  are close to zero. This is also demonstrated in Section 5.

## 4 A Lasso Analog of the GMUS

Rosenbaum and Tsybakov (2010) note that the convex optimization problem

$$\hat{\boldsymbol{\beta}}_{ML} \in \arg \min_{\boldsymbol{\beta}} \frac{1}{2n} \|\mathbf{y} - \mathbf{W}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 + \frac{\delta}{2} \|\boldsymbol{\beta}\|_1^2, \quad (16)$$

which we term the MU lasso (MUL), defines a lasso analog of the MUS. The solutions to (16) are contained in the feasible set of the MUS, i.e.,

$$\hat{\boldsymbol{\beta}}_{ML} \in \left\{ \boldsymbol{\beta} \in \mathbb{R}^p : \frac{1}{n} \left\| \mathbf{W}^T (\mathbf{y} - \mathbf{W}\boldsymbol{\beta}) \right\|_{\infty} \leq \lambda + \delta \|\boldsymbol{\beta}\|_1 \right\},$$



which follows directly from the KKT conditions (Rosenbaum and Tsybakov, 2010). In a similar manner, we can define a generalized MUL (GMUL) by ensuring that its solution  $\hat{\beta}_{ML}^R \in \Theta^R$ , where  $\Theta^R$  is defined in (11). Define  $\hat{\beta}_{ML}^R$  as a vector which satisfies

$$\begin{aligned} \frac{-1}{n} \sum_{i=1}^n w_{ij} \{y_i - \mu(\mathbf{w}_i^T \beta)\} = \\ \tau_j \left\{ \lambda + \sum_{r=1}^R \frac{\delta^r}{r! \sqrt{n}} \|\beta\|_1^r \left\| \mu^{(r)}(\mathbf{W}\beta) \right\|_2 \right\}, \end{aligned} \quad (17)$$

where  $|\tau_j| \leq 1$  and  $\tau_j 1(\beta_j \neq 0) = \text{sign}(\beta_j)$  for  $j = 1, \dots, p$  (Bühlmann and van de Geer, 2011).

We can formulate (17) as a weighted least squares problem. Following (12)-(15), an iterate  $\beta^{(k+1)}$  given the present estimate  $\beta^{(k)}$  is a solution to

$$\frac{-1}{n} \tilde{\mathbf{w}}_j^T (\tilde{\mathbf{z}} - \tilde{\mathbf{W}}\beta) = \tau_j \left\{ \lambda + \sum_{r=1}^R \gamma_r (r+1) \|\beta\|_1^r \right\}, \quad (18)$$

for  $j = 1, \dots, p$ , where

$$\gamma_r = \frac{\delta^r \|\mathbf{V}^{(r)}\|_2}{(r+1)! \sqrt{n}}, \quad r = 1, \dots, R. \quad (19)$$

However, (18) are the KKT conditions corresponding to the convex optimization problem

$$\min. \quad \frac{1}{2n} \left\| \tilde{\mathbf{z}} - \tilde{\mathbf{W}}\beta \right\|_2^2 + \lambda \|\beta\|_1 + \sum_{r=1}^R \gamma_r \|\beta\|_1^{r+1}. \quad (20)$$

Hence, we can find the vector  $\hat{\beta}_{ML}^R$  satisfying (17) by solving the optimization problem (20) in each step of an IRLS algorithm. The convexity of (20) follows from the fact that the composition of the convex L1 norm  $\|\cdot\|_1$  and the convex and nondecreasing power function  $(\cdot)^{r+1}$ ,  $\|\cdot\|_1^{r+1}$ , is itself a convex function (Boyd and Vandenberghe, 2004, p. 84). The solutions will satisfy  $\hat{\beta}_{ML}^R \in \Theta^R$ , and the GMUL can thus be seen as a lasso analog of the GMUS.

Unfortunately, due to the nonseparable penalty terms on the form  $\|\beta\|_1^{r+1}$ , even when  $R = 1$ , (20) is not amenable to fast coordinatewise algorithms which have proven very useful for lasso-type problems (Friedman et al., 2007, 2010). Considering the  $R = 1$  case, it turns out that we can solve a lasso problem at each step of the IRLS algorithm, rather than the challenging problem (20). We start by noting that with  $R = 1$ , (20) is equivalent to

$$\min. \quad \frac{-1}{n} \tilde{\mathbf{z}}^T \tilde{\mathbf{W}}\beta + \beta^T \left\{ \frac{1}{2n} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} + \gamma_1 \mathbf{I}_p \right\} \beta$$

$$+ \lambda \|\beta\|_1 + \gamma_1 \sum_{j=1}^p \sum_{l \neq j} |\beta_j| |\beta_l|.$$

We can now replace the last penalty term in this expression with a weighted L1 penalty. The weight depends on the current estimate  $\beta^{(k)}$ , and will hence be updated at each step of the IRLS algorithm. To be specific, given  $\beta^{(k)}$  we compute

$$\begin{aligned} \beta^{(k+1)} \in \arg \min_{\beta} & \left[ \frac{-1}{n} \tilde{\mathbf{z}}^T \tilde{\mathbf{W}} \beta + \right. \\ & \left. \beta^T \left\{ \frac{1}{2n} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} + \gamma_1 \mathbf{I}_p \right\} \beta + \sum_{j=1}^p \omega_j^{(k)} |\beta_j| \right], \end{aligned} \quad (21)$$

where the weights are given by

$$\omega_j^{(k)} = \lambda + \gamma_1 \sum_{l \neq j} |\beta_l^{(k)}|, \quad j = 1, \dots, p. \quad (22)$$

Since the Hessian  $(2n)^{-1} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} + \gamma_1 \mathbf{I}_p$  is always positive semidefinite for  $\gamma_1 \geq 0$ , (21) is an L1 constrained convex optimization problem, which can be efficiently solved with a coordinate descent algorithm. As we show in Appendix C, the coordinate-wise updates take the form

$$\beta_j \leftarrow \frac{S\left(\frac{1}{n} \sum_{i=1}^n \tilde{w}_{ij} \left(\tilde{z}_i - \sum_{l \neq j} \tilde{w}_{il} \beta_l\right), \omega_j^{(k)}\right)}{\frac{1}{n} \sum_{i=1}^n \tilde{w}_{ij}^2 + 2\gamma_1}, \quad (23)$$

for  $j = 1, \dots, p, 1, \dots$  until convergence, where  $S(\cdot, \cdot)$  denotes the soft-thresholding operator

$$S(a, b) = \begin{cases} a - b, & \text{if } a > 0 \text{ and } b < |a| \\ a + b, & \text{if } a < 0 \text{ and } b < |a| \\ 0, & \text{if } b \geq |a|. \end{cases} \quad (24)$$

We now have the following IRLS algorithm for computing  $\hat{\beta}_{ML}^R$  with  $R = 1$ :

**Require:** An initial estimate  $\beta^{(0)}$  exists,  $k = 1$ .

- 1: **repeat**
- 2:   Compute  $\mathbf{V}^{(1)}$  according to (13)
- 3:   Compute  $\mathbf{z}$  according to (12)
- 4:   Compute  $\tilde{\mathbf{W}}$  and  $\tilde{\mathbf{z}}$  according to (14)
- 5:   Compute  $\gamma_1$  according to (19)
- 6:   Compute  $\omega_j^{(k)}$  according to (22),  $j = 1, \dots, p$
- 7:   Let  $l = 0$  and  $\beta^{(k+1,0)} \leftarrow \beta^{(k)}$
- 8:   **repeat**

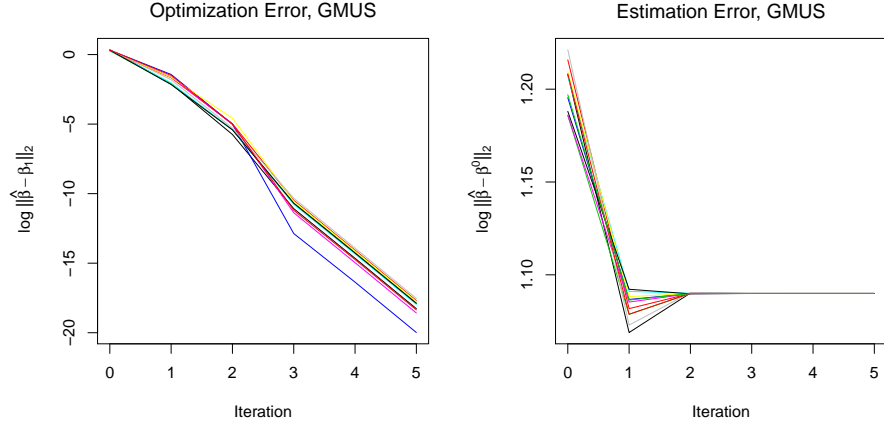


Figure 1: Convergence assessment of the IRLS algorithm for the GMUS.

```

9:   for  $j = 1, \dots, p$  do
10:     Update  $\beta_j^{(k+1, l+1)}$  according to (23)
11:   end for
12:    $l \leftarrow l + 1$ 
13:   until  $\|\beta^{(k+1, l+1)} - \beta^{(k+1, l)}\| < \epsilon_{tol}$ 
14:   return  $\beta^{(k+1)} = \beta^{(k+1, l+1)}$ 
15:    $k \leftarrow k + 1$ 
16: until  $\|\beta^{(k)} - \beta^{(k-1)}\| < \epsilon_{tol}$ 
17: return  $\hat{\beta}_{ML}^R = \beta^{(k)}$ 

```

Again, the algorithm is not guaranteed to converge, because the solution is not in general unique, but we have experienced good convergence properties in practice as long as the constraint parameters  $\lambda$  and  $\delta$  are not too small. See also Section 5.

## 5 Convergence of IRLS Algorithms

We tested the convergence of the IRLS algorithm for the GMUS with  $R = 1$  using a technique similar to, e.g., Loh and Wainwright (2012). We generated a problem instance with  $n = 200$ ,  $p = 500$ ,  $s = 10$ , and  $\beta^0 = (1, \dots, 1, 0, \dots, 0)^T$ . The matrix of covariates had standard normally distributed entries  $x_{ij} \sim N(0, 1)$  and the measurement matrix had entries  $w_{ij} = x_{ij} + u_{ij}$ , with  $u_{ij} \sim N(0, \sigma_u)$ , for  $i = 1, \dots, n$  and  $j = 1, \dots, p$ , with  $\sigma_u = 0.2$ . The response was binomially distributed with mean  $(1 + \exp\{-\mathbf{x}_i^T \beta^0\})^{-1}$ , and we set  $\lambda = (1/3)\sqrt{\log p/n}$  and  $\delta = 0.1$ . The value of  $\lambda$  was chosen to be on the same order of magnitude as the theoretically optimal value, cf. (Bühlmann and van de Geer, 2011, p. 127). The GMUS with a logistic link function was computed 11 times, each time with a random starting point for the IRLS algorithm. The left plot in Figure 1 shows

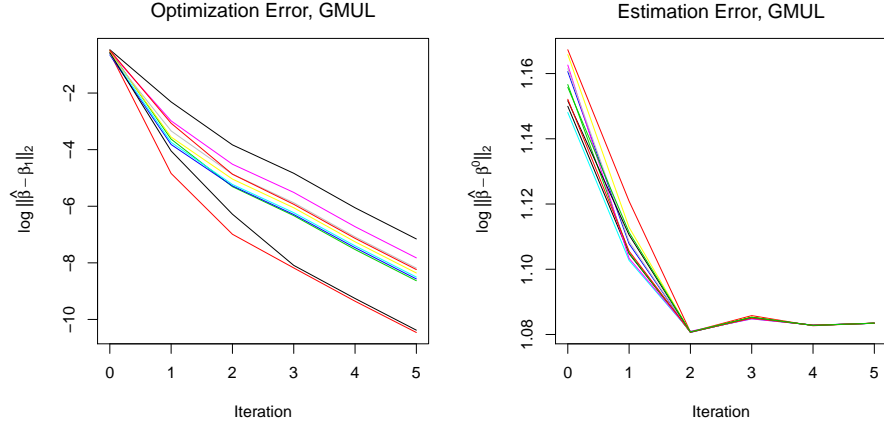


Figure 2: Convergence assessment of the IRLS algorithm for the GMUL.

optimization errors of the last ten runs computed as the logarithm of the L2 distance between the current iterate and the corresponding estimate obtained in the first run. This gives a picture of the sensitivity to initial conditions, and we see that the difference between the estimates gets extremely small after very few iterations. The right plot in Figure 1 shows the statistical error computed as the L2 distance between the current iterate and  $\beta^0$  for each of the 11 runs. Here we see that after two iterations of the IRLS algorithm, the statistical error is indistinguishable between the different runs. The algorithm converged very quickly in this case. We experienced the same good convergence properties in the simulation experiments of Section 6 and on the microarray data of Section 7.

The same experiment was done for the IRLS algorithm for the GMUL, and the results are shown in Figure 2. Also here we see that the algorithm converges with very few iterations. For the GMUL, the inner coordinate descent algorithm converged in less than ten steps in every iteration.

## 6 Simulation Experiments

In this section we describe simulation experiments comparing the GMUL and the GMUS to the standard formulations of the GDS and the lasso. In all cases considered, we set  $n = 200$  and  $s = 10$ . The matrix  $\mathbf{X}$  had i.i.d. entries  $x_{ij} \sim N(0, 1)$ , and conditional on  $\mathbf{X}$ ,  $\mathbf{W}$  had i.i.d. entries  $w_{ij} = x_{ij} + u_{ij}$  with  $u_{ij} \sim N(0, \sigma_u)$ , for  $i = 1, \dots, n$  and  $j = 1, \dots, p$ . The response  $y_i$  was either binomially distributed with mean  $(1 + \exp\{-\mathbf{x}_i^T \beta^0\})^{-1}$  or Poisson distributed with mean  $\exp(\mathbf{x}_i^T \beta^0)$ . For each problem instance with data  $(\mathbf{W}, \mathbf{y})$ , two GDS and lasso fits were computed via ten-fold cross-validation: one fit corresponding to the minimum cross validated deviance, whose regularization parameter we

denote  $\hat{\lambda}_{min}$ , and one fit corresponding to the largest regularization parameter within one standard error of the minimum, whose regularization parameter we denote  $\hat{\lambda}_{se}$ . The latter is suggested in (Hastie et al., 2009, p. 244). The solution to the GMUL was computed over a discrete grid of  $\delta$  values, fixing  $\lambda$  at the  $\hat{\lambda}_{min}$  or  $\hat{\lambda}_{se}$  obtained in the lasso fit. The same procedure was performed for the GMUS, using  $\hat{\lambda}_{min}$  or  $\hat{\lambda}_{se}$  from the GDS fit. For each setting, this procedure was repeated in 100 independent Monte Carlo experiments.

Finally, the Monte Carlo average of the number of nonzero coefficients was plotted against the value of  $\delta$ , as shown in Figure 3. According to the *elbow rule* (Rosenbaum and Tsybakov, 2010),  $\delta$  was chosen where the curve begins to flatten. Hence, a common regularization parameter  $\delta$  was chosen for all Monte Carlo simulations, while the  $\lambda$  was chosen by cross-validation separately in each case. In Section 7 we will show a real application, where  $\delta$  is of course chosen specifically for the data set at hand.

The lasso solution was computed using the R package `glmnet` (Friedman et al., 2010) and the GMUL was implemented in R and C++ using the `Rcpp` package (Eddelbuettel and François, 2011; Eddelbuettel, 2013). The GDS and GMUS were implemented in R, utilizing the high-level interface to the GNU Linear Programming Kit provided by the package `Rglpk` (Theussl and Hornik, 2013) to compute the solution to the linear program (15) in each step of the IRLS algorithm.

## 6.1 Logistic Regression

For the data with binomially distributed response, the logistic regression versions of the methods were used. The nonzero regression coefficients were set to  $\beta_j^0 = 1$  for  $j = 1, \dots, s$ , and the number of covariates was  $p = 500$ . The measurement error standard deviation was either  $\sigma_u = 0.2$  or  $\sigma_u = 0.5$ , respectively.

Figure 3 shows the elbow plot for the GMUS and GDS in the  $\sigma_u = 0.2$  case. We see that the elbows are quite well defined. The elbow plots for all other settings are shown in the supplementary material.

Table 1 shows the covariate selection properties of the different methods in the  $\sigma_u = 0.2$  case, with regularization parameters set as shown in Figure 3. Overall, the GMUL and the GMUS display a drastic reduction in the number of false positives (FPs) compared to the GDS and the lasso, at the cost of a slight reduction in the number of true positives (TPs). This is also evident in the precision, the number of relevant covariates among those selected, shown in the rightmost column. While the lasso and GDS estimates computed have precision ranging from 0.19 to 0.43, the GMUL and the GMUS estimates have precision from 0.50 up to 0.63. Similar to the conclusion of Rosenbaum and Tsybakov (2010) for linear models, Table 1 shows that the standard formulations of the lasso and the GDS for logistic regression select too many covariates in the presence of measurement error. The GMUL and GMUS, which take the measurement error into account through the additional regularization parameter  $\delta$ , sharply improve the precision.

Table 2 shows the similar results with  $\sigma_u = 0.5$ . This is a harder problem,

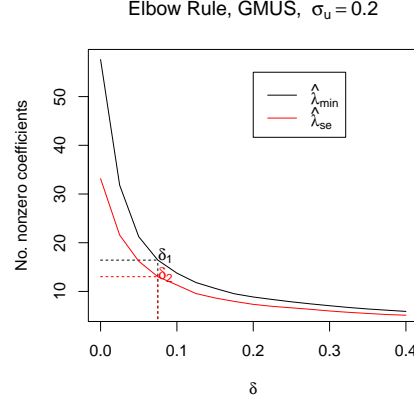


Figure 3: Elbow rule with logistic regression. The average number of nonzero coefficients is plotted against  $\delta$ .  $\hat{\lambda}_{min}$  denotes the value of  $\lambda$  minimizing the cross-validation error for the GDS while  $\hat{\lambda}_{se}$  denotes the largest  $\lambda$  within one standard error of the minimum.  $\delta = 0$  corresponds to the GDS.

Table 1: Results from the simulation experiment with logistic regression, with  $p = 500$  and  $\sigma_u = 0.2$ . #TP denotes the number of true positives and #FP the number of false positives. Standard errors are shown in parentheses.

	#TP	#FP	#TP/(#TP + #FP)
Lasso( $\hat{\lambda}_{min}$ )	9.64 (0.07)	40.42 (1.59)	0.22 (0.01)
GMUL( $\hat{\lambda}_{min}, \delta_1$ )	8.56 (0.17)	10.14 (0.65)	0.50 (0.01)
Lasso( $\hat{\lambda}_{se}$ )	8.93 (0.13)	15.15 (0.91)	0.43 (0.02)
GMUL( $\hat{\lambda}_{se}, \delta_2$ )	8.57 (0.15)	8.24 (0.45)	0.55 (0.01)
GDS( $\hat{\lambda}_{min}$ )	9.74 (0.05)	47.87 (1.71)	0.19 (0.01)
GMUS( $\hat{\lambda}_{min}, \delta_1$ )	8.54 (0.11)	7.89 (0.36)	0.54 (0.01)
GDS( $\hat{\lambda}_{se}$ )	9.30 (0.10)	23.86 (1.30)	0.32 (0.01)
GMUS( $\hat{\lambda}_{se}, \delta_2$ )	7.79 (0.13)	5.24 (0.31)	0.63 (0.02)

Table 2: Results from the simulation experiment with logistic regression, with  $p = 500$  and  $\sigma_u = 0.5$ . #TP denotes the number of true positives and #FP the number of false positives. Standard errors are shown in parentheses.

	#TP	#FP	#TP/(#TP + #FP)
Lasso( $\hat{\lambda}_{min}$ )	8.54 (0.16)	31.68 (1.74)	0.26 (0.01)
GMUL( $\hat{\lambda}_{min}, \delta_1$ )	7.42 (0.18)	11.72 (0.68)	0.44 (0.02)
Lasso( $\hat{\lambda}_{se}$ )	7.11 (0.22)	11.63 (1.03)	0.49 (0.02)
GMUL( $\hat{\lambda}_{se}, \delta_2$ )	6.47 (0.22)	7.09 (0.61)	0.57 (0.02)
GDS( $\hat{\lambda}_{min}$ )	8.95 (0.14)	43.45 (2.03)	0.20 (0.01)
GMUS( $\hat{\lambda}_{min}, \delta_1$ )	7.72 (0.17)	9.73 (0.52)	0.48 (0.01)
GDS( $\hat{\lambda}_{se}$ )	8.04 (0.19)	19.90 (1.50)	0.37 (0.02)
GMUS( $\hat{\lambda}_{se}, \delta_2$ )	7.16 (0.18)	7.76 (0.55)	0.54 (0.02)

as the measurement error is rather large, but also in this case the GMUL and the GMUS reduce the number of FPs considerably.

## 6.2 Poisson Regression

For the data with Poisson distributed response, the Poisson regression versions of the methods were used. The nonzero regression coefficients were set to  $\beta_j^0 = 0.2$  for  $j = 1, \dots, s$ , and the measurement error standard deviation was  $\sigma_u = 0.2$ .

Since the exponential mean function for Poisson regression is more highly nonlinear than the mean function for logistic regression, one might suspect that the first order Taylor approximation considered in this paper yields a poorer fit for Poisson regression. We therefore started with  $p = 150$  covariates, creating a somewhat easier problem. Table 3 shows the results, from which it is clear that the standard lasso and GDS select too many covariates. First of all, the GDS and lasso with  $\lambda = \hat{\lambda}_{min}$  select a large number of false positives. Secondly, the GMUL and the GMUS with  $\lambda = \hat{\lambda}_{min}$  perform better than the GDS and lasso with  $\lambda = \hat{\lambda}_{se}$ . For example, the GMUL with  $\lambda = \hat{\lambda}_{min}$  selects on average 9.51 out of the 10 TPs and 6.97 FPs. The lasso with  $\lambda = \hat{\lambda}_{se}$ , on the other hand, selects 9.65 TPs and 10.64 FPs. Hence, while the number of TPs selected is almost identical, the lasso selects on average 3.67 more FPs than the GMUS in this case. The corresponding numbers for the GMUS are 9.34 TPs and 12.12 FPs, versus 9.44 TPs and 16.87 FPs of the GDS.

Table 4 shows the corresponding results with  $p = 500$ . For some of the methods, there were cases in which no covariate was selected. In these cases, the average precision and its standard error are not well defined, and we report instead the average number of TPs divided by the average number of selected covariates, and mark the corresponding cells with an asterisk. That is, the numbers marked with an asterisk in the fourth column were computed by dividing the number in the second column in the same row by the sum of the numbers

Table 3: Results from the simulation experiment with Poisson regression, with  $p = 150$  and  $\sigma_u = 0.2$ . #TP denotes the number of true positives and #FP the number of false positives. Standard errors are shown in parentheses.

	#TP	#FP	#TP/(#TP + #FP)
Lasso( $\hat{\lambda}_{min}$ )	9.97 (0.02)	28.09 (1.07)	0.28 (0.01)
GMUL( $\hat{\lambda}_{min}, \delta_1$ )	9.51 (0.06)	6.97 (0.30)	0.59 (0.01)
Lasso( $\hat{\lambda}_{se}$ )	9.65 (0.07)	10.64 (0.65)	0.52 (0.02)
GMUL( $\hat{\lambda}_{se}, \delta_2$ )	9.37 (0.09)	6.48 (0.41)	0.63 (0.01)
GDS( $\hat{\lambda}_{min}$ )	9.95 (0.02)	43.44 (1.54)	0.20 (0.01)
GMUS( $\hat{\lambda}_{min}, \delta_1$ )	9.34 (0.09)	12.12 (0.53)	0.46 (0.01)
GDS( $\hat{\lambda}_{se}$ )	9.44 (0.09)	16.87 (1.06)	0.41 (0.01)
GMUS( $\hat{\lambda}_{se}, \delta_2$ )	7.71 (0.16)	3.51 (0.29)	0.73 (0.02)

Table 4: Results from the simulation experiment with Poisson regression, with  $p = 500$  and  $\sigma_u = 0.2$ . #TP denotes the number of true positives and #FP the number of false positives. Standard errors are shown in parentheses.

	#TP	#FP	#TP/(#TP + #FP)
Lasso( $\hat{\lambda}_{min}$ )	9.80 (0.05)	42.54 (1.94)	0.21 (0.01)
GMUL( $\hat{\lambda}_{min}, \delta_1$ )	9.14 (0.10)	13.94 (0.56)	0.42 (0.01)
Lasso( $\hat{\lambda}_{se}$ )	8.23 (0.23)	12.00 (0.81)	0.41*
GMUL( $\hat{\lambda}_{se}, \delta_2$ )	7.97 (0.23)	8.27 (0.54)	0.49*
GDS( $\hat{\lambda}_{min}$ )	9.58 (0.12)	57.96 (2.15)	0.14*
GMUS( $\hat{\lambda}_{min}, \delta_1$ )	8.01 (0.15)	12.60 (0.54)	0.39*
GDS( $\hat{\lambda}_{se}$ )	8.24 (0.19)	22.88 (1.64)	0.26*
GMUS( $\hat{\lambda}_{se}, \delta_2$ )	7.03 (0.20)	8.41 (0.67)	0.46*



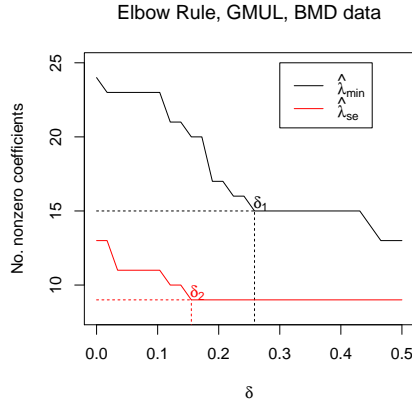


Figure 4: Elbow rule for the BMD example.

in the second and third columns in the same row. The GMUS and GMUL have consistently better covariate selection performance than the lasso and the GDS. For example, comparing Lasso( $\hat{\lambda}_{min}$ ) and GMUL( $\hat{\lambda}_{min}, \delta_1$ ), we see that adding the additional regularization  $\delta_1$  after cross-validation doubles the precision, by reducing the average number of irrelevant covariates selected by 28.6.

## 7 Analysis of a Microarray Data Set

We now show an application to a data set containing 22,815 normalized microarray gene expression measurements as well as measurements of bone mineral density (BMD), for 84 Norwegian women (Rippe et al., 2010). As microarray measurements are known to be noisy and subject to various sources of bias (Boulesteix et al., 2008; Rocke and Durbin, 2001; Tadesse et al., 2005), this context is very appropriate for the GMUL and the GMUS. The total hip T-score BMD was used as outcome variable, and the subjects were classified as having BMD below median ( $y = 1$ ) or above median ( $y = 0$ ).

Our IRLS algorithm, with an inner coordinate descent loop for computing the GMUL estimate, scales very well with the number of covariates  $p$ , and we were therefore able to compute GMUL estimates for the full dataset with  $p = 22,815$ . The current implementation of the GMUS, on the other hand, is based on using the simplex method to solve the linear program (15) in each step of the IRLS algorithm. Doing this repeatedly over a grid of  $\lambda$  and  $\delta$  values, turned out to be computationally infeasible when  $p$  is this large. We thus only report results obtained using the lasso and GMUL. Developing a tailor-made solver for (15) using the interior-point method (Boyd and Vandenberghe, 2004) could overcome this problem, but is beyond the scope of this paper.

As in the simulation experiments, we computed two standard lasso fits of

the BMD data, using `glmnet` (Friedman et al., 2010): One corresponding to the minimum cross-validated deviance, with regularization parameter  $\hat{\lambda}_{min}$ , and one corresponding to  $\hat{\lambda}_{se}$ , the largest regularization parameter within one standard error of the minimum cross-validated deviance. Starting at these two  $\lambda$  values, we gradually increased  $\delta$  from 0 on a discrete grid with spacing 0.025. Figure 4 shows the elbow plots for these two GMUL estimators. Both curves have a long range of  $\delta$  values for which the number of nonzero coefficients is constant, and we choose the corresponding regularization parameters  $\delta_1$  and  $\delta_2$  at the minimum values over these regions.

Table 5 shows the gene symbols and standardized regression coefficients for the lasso estimate with  $\lambda = \hat{\lambda}_{min}$  and the GMUL with  $\lambda = \hat{\lambda}_{min}$  and  $\delta = \delta_1$ . The lasso selected 23 genes and one control sequence, and the GMUL reduced this number to  $13 + 1$ , but all genes selected by the GMUL were also selected by the lasso. The GMUL hence worked by removing genes from the selected set, without adding any new ones. Based on our simulation experiments, it seems plausible that the genes selected by the lasso but not by the GMUL were indeed false positives.

Table 6 shows the corresponding table when using  $\lambda = \hat{\lambda}_{se}$ . Here, the lasso selected 12 genes and one control sequence, all of which were also chosen by the lasso with  $\lambda = \hat{\lambda}_{min}$ . The GMUL removed 5 genes, ending up with  $7 + 1$  selected covariates. Again, the genes selected by the GMUL were a subset of the genes selected by the lasso. The genes selected by the GMUL with  $\lambda = \hat{\lambda}_{se}$  were also a subset of the genes selected by the GMUL with  $\lambda = \hat{\lambda}_{min}$ .

## 8 Conclusion

This paper focuses on covariate selection in high-dimensional GLMs when the covariates are subject to measurement error. We generalize the MUS (Rosenbaum and Tsybakov, 2010), which is limited to linear models, by considering an  $R$ th order Taylor approximation of the GLM mean function. Furthermore, we develop the GMUL, a lasso analog of the GMUS. By not requiring an estimate of the measurement error covariance matrix, the proposed methods give an important practical advantage.

Both the GMUS and the GMUL can be computed using IRLS. For computational reasons, we only consider Taylor approximations of order  $R = 1$ . In this case, the GMUS requires solving a linear program at each step of the IRLS algorithm, while the GMUL can be computed with an inner coordinate descent loop at each step of the IRLS algorithm. We demonstrate in simulation experiments with logistic and Poisson regression, that the standard formulations of the lasso and GDS select a large number of false positives when the covariates are subject to measurement error, and that the GMUS and the GMUL can reduce this number while mainly keeping the true positives.

As the main focus of this paper is covariate selection in the presence of measurement error, the simulation results in Section 6 focused on the number of TPs, FPs, and the precision. Another important aspect is the estimation

Table 5: Gene symbols and standardized regression coefficients of genes selected by lasso and GMUL with  $\lambda = \hat{\lambda}_{min}$ .

Lasso( $\hat{\lambda}_{min}$ )		GMUL( $\hat{\lambda}_{min}, \delta_1$ )	
Gene symbol	Coef.	Probe ID	Coef.
MIR22HG	0.36	MIR22HG	0.23
FAM118B	0.32	FAM118B	0.20
ITGBL1	-0.22	BE551142	-0.18
(control seq.)	-0.21	(control seq.)	-0.15
BE551142	-0.20	ADI1	0.12
FKBP5	0.16	POLR2D	-0.09
ADI1	0.15	ITGBL1	-0.09
Hs.380027	-0.14	PCGF1	-0.09
POLR2D	-0.12	SOST	-0.09
PCGF1	-0.10	COPS4	0.06
RPH3A	-0.10	FKBP5	0.04
COPS4	0.10	ELTD1	0.04
Hs.633128	-0.09	CARD8	-0.02
TSC22D1-AS1	-0.08	Hs.633128	-0.02
CARD8	-0.08		
COL11A1	-0.08		
AA463449	-0.08		
CCDC169	-0.07		
SOST	-0.06		
AF086063	-0.06		
MON1B	-0.06		
DYNLRB1	-0.05		
ELTD1	0.04		
SEMA3F	0.00		

Table 6: Gene symbols and standardized regression coefficients of genes selected by lasso and GMUL with  $\lambda = \hat{\lambda}_{se}$ .

Lasso( $\hat{\lambda}_{min}$ )		GMUL( $\hat{\lambda}_{min}, \delta_1$ )	
Gene symbol	Coef.	Gene symbol	Coef.
MIR22HG	0.26	MIR22HG	0.21
FAM118B	0.23	FAM118B	0.19
BE551142	-0.18	BE551142	-0.14
(control seq.)	-0.14	(control seq.)	-0.12
POLR2D	-0.10	POLR2D	-0.08
ADI1	0.09	ADI1	0.05
SOST	-0.07	SOST	-0.05
PCGF1	-0.07	PCGF1	-0.04
ITGBL1	-0.04		
COPS4	0.02		
ELTD1	0.01		
CARD8	-0.01		
Hs.633128	-0.01		

error. We have observed in practice that the GMUL and the GMUS tend to have a smaller  $\ell_1$  estimation error than the lasso and GDS, and a larger  $\ell_2$  estimation error. This can be explained by the fact that the GMUL and GMUS have better covariate selection properties (lower  $\ell_1$  error), and penalize more (higher  $\ell_2$  error). If a low  $\ell_2$  estimation error is of main interest, it may therefore be wiser to use the bias correction proposed by Loh and Wainwright (2012) or Rosenbaum and Tsybakov (2013).

In our current implementations, the GMUL can efficiently handle  $p$  on the order of tens of thousands, while the linear program (15) solved in each step of the IRLS algorithm for the GMUS becomes increasingly slow when  $p$  is larger than about a thousand. An interesting problem for further study is therefore to develop an algorithm for (15) which scales better with  $p$ . In addition, our current implementations are restricted to first order Taylor approximations. Although this has been shown to work well, it would be interesting to develop algorithms for computing the GMUL and the GMUS using higher order approximations.

Although not considered in this paper, the GMUL and the GMUS can also be used in problems with missing data. This is particularly relevant when imputation methods are computationally infeasible due to the large number of covariates.

## A Proof of Proposition 1

It follows from the Taylor series expansion (9) that

$$\mu(\mathbf{w}_i^T \boldsymbol{\beta}^0) = \mu(\mathbf{x}_i^T \boldsymbol{\beta}^0) - \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r.$$

This gives, for  $j = 1, \dots, p$ ,

$$\begin{aligned} & \left| \frac{1}{n} \sum_{i=1}^n w_{ij} (y_i - \mu(\mathbf{w}_i^T \boldsymbol{\beta}^0)) \right| = \\ & \left| \frac{1}{n} \sum_{i=1}^n w_{ij} \left( \epsilon_i + \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r \right) \right| \leq \\ & \left| \frac{1}{n} \sum_{i=1}^n w_{ij} \epsilon_i \right| + \left| \frac{1}{n} \sum_{i=1}^n w_{ij} \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r \right| \leq \\ & \lambda + \left| \frac{1}{n} \sum_{i=1}^n w_{ij} \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r \right| \leq \\ & \lambda + \left| \frac{1}{n} \sum_{i=1}^n w_{ij} \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r \right|, \end{aligned}$$

where we inserted  $\epsilon_i = y_i - \mu(\mathbf{w}_i^T \boldsymbol{\beta}^0)$  in the first step, we used the triangle inequality in the second step, we inserted the left bound in (7) in the third step, and finally used the generalized triangle inequality. Next, we have

$$\begin{aligned} & \left| \frac{1}{n} \sum_{i=1}^n w_{ij} \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r \right| \leq \\ & \frac{1}{n} \left( \sum_{i=1}^n w_{ij}^2 \right)^{\frac{1}{2}} \left( \sum_{i=1}^n \left\{ \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r \right\}^2 \right)^{\frac{1}{2}} \\ & = \frac{1}{\sqrt{n}} \left( \sum_{i=1}^n \left\{ \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r \right\}^2 \right)^{\frac{1}{2}}, \end{aligned}$$

where we used Hölder's inequality in the first step and the assumption (5) that the covariates are standardized to have mean zero and unit variance in the second step. We now note that the last term above is the L2 norm  $\|\mathbf{v}\|_2$  of a vector  $\mathbf{v} \in \mathbb{R}^n$  with elements

$$v_i = \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r, \quad i = 1, \dots, n.$$

We thus have

$$\begin{aligned}
& \frac{1}{\sqrt{n}} \left( \sum_{i=1}^n \left\{ \sum_{r=1}^{\infty} \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^r \right\}^2 \right)^{\frac{1}{2}} \leq \\
& \frac{1}{\sqrt{n}} \sum_{r=1}^{\infty} \left( \sum_{i=1}^n \left\{ \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} \right\}^2 (-\mathbf{u}_i^T \boldsymbol{\beta}^0)^{2r} \right)^{\frac{1}{2}} \\
& \frac{1}{\sqrt{n}} \sum_{r=1}^{\infty} \left( \sum_{i=1}^n \left\{ \frac{\mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0)}{r!} \right\}^2 \|\mathbf{u}_i\|_{\infty}^{2r} \|\boldsymbol{\beta}\|_1^{2r} \right)^{\frac{1}{2}} \leq \\
& \sum_{r=1}^{\infty} \frac{\delta^r \|\boldsymbol{\beta}\|_1^r}{r! \sqrt{n}} \left( \sum_{i=1}^n \left\{ \mu^{(r)}(\mathbf{w}_i^T \boldsymbol{\beta}^0) \right\}^2 \right)^{\frac{1}{2}} = \\
& \sum_{r=1}^{\infty} \frac{\delta^r \|\boldsymbol{\beta}\|_1^r}{r! \sqrt{n}} \left\| \boldsymbol{\mu}^{(r)}(\mathbf{W} \boldsymbol{\beta}^0) \right\|_2,
\end{aligned}$$

where we used the triangle inequality in the first step, Höfdding's inequality in the second step, and finally used the right bound in (7) in the second last step.

Putting the pieces together, it follows that

$$\frac{1}{n} \left| \sum_{i=1}^n w_{ij} (y_i - \mu(\mathbf{w}_i^T \boldsymbol{\beta}^0)) \right| \leq \lambda + \sum_{r=1}^{\infty} \frac{\delta^r}{r! \sqrt{n}} \|\boldsymbol{\beta}^0\|_1^r \left\| \boldsymbol{\mu}^{(r)}(\mathbf{W} \boldsymbol{\beta}^0) \right\|_2$$

for  $j = 1, \dots, p$ , which proves that  $\boldsymbol{\beta}^0 \in \Theta$ .

## B Computing the Solution to (15)

We can simplify the computation of (15) by introducing the auxiliary variable  $\mathbf{u} \in \mathbb{R}^p$ . We then get the equivalent problem

$$\begin{aligned}
& \text{minimize } \mathbf{1}_p^T \mathbf{u} \text{ (with respect to } \mathbf{u}, \boldsymbol{\beta}) \\
& \text{subject to } -\mathbf{u} \leq \boldsymbol{\beta} \leq \mathbf{u}, \\
& -\sum_{r=1}^R \frac{\delta^r}{r! \sqrt{n}} (\mathbf{1}_p^T \mathbf{u})^r \left\| \mathbf{V}^{(r)} \right\|_2 \mathbf{1}_p + \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \boldsymbol{\beta} \leq \lambda \mathbf{1}_p + \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{z}}, \\
& \text{and} \\
& -\sum_{r=1}^R \frac{\delta^r}{r! \sqrt{n}} (\mathbf{1}_p^T \mathbf{u})^r \left\| \mathbf{V}^{(r)} \right\|_2 \mathbf{1}_p - \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \boldsymbol{\beta} \leq \lambda \mathbf{1}_p - \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{z}}.
\end{aligned}$$

When  $R = 1$ , the problem (15) is thus equivalent to the linear program

$$\text{minimize } \mathbf{1}_p^T \mathbf{u} \text{ (with respect to } \mathbf{u}, \boldsymbol{\beta})$$

$$\begin{aligned}
& \text{subject to } -\mathbf{u} \leq \boldsymbol{\beta} \leq \mathbf{u}, \\
& -\frac{\delta}{\sqrt{n}} \mathbf{1}_p^T \mathbf{u} \left\| \mathbf{V}^{(1)} \right\|_2 \mathbf{1}_p + \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \boldsymbol{\beta} \leq \lambda \mathbf{1}_p + \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{z}}, \\
& \text{and} \\
& -\frac{\delta}{\sqrt{n}} \mathbf{1}_p^T \mathbf{u} \left\| \mathbf{V}^{(1)} \right\|_2 \mathbf{1}_p - \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \boldsymbol{\beta} \leq \lambda \mathbf{1}_p - \frac{1}{n} \tilde{\mathbf{W}}^T \tilde{\mathbf{z}},
\end{aligned}$$

which can be solved by standard software.

## C Coordinate Descent Algorithm for GMUL

We describe here the coordinate descent algorithm used to solve (21). Our goal is to find a  $\boldsymbol{\beta}$  minimizing the function

$$f(\boldsymbol{\beta}) = -n^{-1} \tilde{\mathbf{z}}^T \tilde{\mathbf{W}} \boldsymbol{\beta} + \boldsymbol{\beta}^T \left\{ (2n)^{-1} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} + \gamma_1 \mathbf{I}_p \right\} \boldsymbol{\beta} + \sum_{j=1}^p \omega_j^{(k)} |\beta_j|,$$

which can be written equivalently as

$$\begin{aligned}
f(\boldsymbol{\beta}) = & -n^{-1} \sum_{i=1}^n \tilde{z}_i \sum_{j=1}^p \tilde{w}_{ij} \beta_j + (2n)^{-1} \sum_{i=1}^n \left( \sum_{j=1}^p \tilde{w}_{ij} \beta_j \right)^2 \\
& + \gamma_1 \sum_{j=1}^p \beta_j^2 + \sum_{j=1}^p \omega_j^{(k)} |\beta_j|.
\end{aligned}$$

The partial derivatives of  $f(\boldsymbol{\beta})$  with respect to  $\beta_j$ ,  $j = 1, \dots, p$ , can be written as

$$\begin{aligned}
\frac{\partial f}{\partial \beta_j} = & -n^{-1} \sum_{i=1}^n \tilde{z}_i \tilde{w}_{ij} + n^{-1} \sum_{i=1}^n \tilde{w}_{ij} \sum_{l \neq j} \tilde{w}_{il} \beta_l \\
& + n^{-1} \beta_j \sum_{i=1}^n \tilde{w}_{ij}^2 + 2\gamma_1 \beta_j + \omega_j^{(k)} \tau_j,
\end{aligned}$$

where  $\tau_j = 1$  if  $\beta_j > 0$ ,  $\tau_j = -1$  if  $\beta_j < 0$ , and  $\tau_j \in [-1, 1]$  if  $\beta_j = 0$ . Setting  $\partial f / \partial \beta_j = 0$ , we find the analytical solution to (21),

$$\beta_j = \frac{n^{-1} \sum_{i=1}^n \tilde{w}_{ij} \left( \tilde{z}_i - \sum_{l \neq j} \tilde{w}_{il} \beta_l \right) - \omega_j^{(k)} \tau_j}{n^{-1} \sum_{i=1}^n \tilde{w}_{ij}^2 + 2\gamma_1}$$

for  $j = 1, \dots, p$ . Since  $\boldsymbol{\tau}$  is implicitly defined, we compute  $\boldsymbol{\beta}$  iteratively using the coordinate descent updates

$$\hat{\beta}_j \leftarrow \frac{S \left( n^{-1} \sum_{i=1}^n \tilde{w}_{ij} \left( \tilde{z}_i - \sum_{l \neq j} \tilde{w}_{il} \hat{\beta}_l \right), \omega_j^{(k)} \right)}{\frac{1}{n} \sum_{i=1}^n \tilde{w}_{ij}^2 + 2\gamma_1},$$

for  $j = 1, \dots, p, 1, \dots$  until convergence (Friedman et al., 2007).  $S(a, b)$  is the soft-thresholding operator (24). On convergence, we set  $\beta^{(k+1)} = \hat{\beta}$ .

## Acknowledgment

The authors would like to thank Bin Yu for discussions and Sjur Reppe for providing the bone density data.

## References

- Antoniadis, A., P. Fryzlewicz, and F. Letue (2010). The Dantzig selector in Cox’s proportional hazards model. *Scandinavian Journal of Statistics* 37, 531–552.
- Bertrand, A. and M. Moonen (2011, May). Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks. *Signal Processing, IEEE Transactions on* 59(5), 2320–2330.
- Bickel, P. J., Y. Ritov, and A. B. Tsybakov (2009). Simultaneous analysis of lasso and Dantzig selector. *Annals of Statistics* 37(4), 1705–1732.
- Boulesteix, A.-L., C. Strobl, T. Augustin, and M. Daumer (2008). Evaluating microarray-based classifiers: An overview. *Cancer Informatics* 6.
- Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. New York, USA: Cambridge University Press.
- Bühlmann, P. and S. van de Geer (2011). *Statistics for high-dimensional data*. Berlin, Heidelberg, Germany: Springer.
- Candes, E. and T. Tao (2007). The Dantzig selector: statistical estimation when  $p$  is much larger than  $n$ . *Annals of Statistics* 35(6), 2313–2351.
- Carroll, R. J., D. Ruppert, L. A. Stefanski, and C. M. Crainiceanu (2006). *Measurement Error in Nonlinear Models*. Boca Raton, Florida, USA: Chapman & Hall/CRC.
- Chen, Y. and C. Caramanis (2013). Noisy and missing data regression: Distribution-oblivious support recovery. In *JMLR Workshop and Conference Proceedings*, Volume 28, pp. 381–391.
- Eddelbuettel, D. (2013). *Seamless R and C++ Integration with Rcpp*. New York, NY: Springer.
- Eddelbuettel, D. and R. François (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software* 40(8), 1–18.
- Friedman, J., T. Hastie, H. Höfling, and R. Tibshirani (2007). Pathwise coordinate optimization. *Annals of Applied Statistics* 1(2), 302–332.



- Friedman, J. H., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33(1), 1–22.
- Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*. Springer.
- James, G. M. and P. Radchenko (2009). A generalized Dantzig selector with shrinkage tuning. *Biometrika* 96(2), 323–337.
- Kipnis, V., A. F. Subar, D. Midthune, L. S. Freedman, R. Ballard-Barbash, R. P. Troiano, S. Bingham, D. A. Schoeller, A. Schatzkin, and R. J. Carroll (2003). Structure of dietary measurement error: Results of the open biomarker study. *American Journal of Epidemiology* 158(1), 14–21.
- Liang, H. and R. Li (2009). Variable selection for partially linear models with measurement errors. *Journal of the American Statistical Association* 104(485), 234–248.
- Loh, P.-L. and M. J. Wainwright (2012). High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *Annals of Statistics* 40(3), 1637–1664.
- Ma, Y. and R. Li (2010). Variable selection in measurement error models. *Bernoulli* 16(1), 274–300.
- McCullagh, P. and J. Nelder (1989). *Generalized Linear Models*. Boca Raton, Florida, USA: Chapman & Hall / CRC.
- Meinshausen, N. and P. Bühlmann (2006). High-dimensional graphs and variable selection with the lasso. *Annals of Statistics* 34(3), pp. 1436–1462.
- Nguyen, N. and T. Tran (2013, April). Robust lasso with missing and grossly corrupted observations. *Information Theory, IEEE Transactions on* 59(4), 2036–2058.
- Reppe, S., H. Refvem, V. T. Gautvik, O. K. Olstad, P. I. Høvring, F. P. Reinholdt, M. Holden, A. Frigessi, R. Jemtland, and K. M. Gautvik (2010). Eight genes are highly associated with BMD variation in postmenopausal Caucasian women. *Bone* 46, 604–612.
- Rocke, D. M. and B. Durbin (2001). A model for measurement error for gene expression arrays. *Journal of Computational Biology* 8.
- Rosenbaum, M. and A. B. Tsybakov (2010). Sparse recovery under matrix uncertainty. *Annals of Statistics* 38(5), 2620–2651.
- Rosenbaum, M. and A. B. Tsybakov (2013). Improved matrix uncertainty selector. In *From Probability to Statistics and Back: High-Dimensional Models and Processes – A Festschrift in Honor of Jon A. Wellner*, pp. 276 – 290. Beachwood, Ohio, USA: Institute of Mathematical Statistics.

- Sørensen, Ø., A. Frigessi, and M. Thoresen (2014). Measurement error in lasso: Impact and likelihood bias correction. *arXiv:1210.5378 [stat.ME]*. to appear in *Statistica Sinica*.
- Tadesse, M. G., J. G. Ibrahim, R. Gentleman, S. Chiaretti, J. Ritz, and R. Foa (2005). Bayesian error-in-variable survival model for the analysis of genechip arrays. *Biometrics* 61, 488–497.
- Theussl, S. and K. Hornik (2013). *Rglpk: R/GNU Linear Programming Kit Interface*. R package version 0.5-2.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B* 58(1), 267–288.
- van de Geer, S. A. (2008). High-dimensional generalized linear models and the lasso. *Annals of Statistics* 36(2), pp. 614–645.
- van de Geer, S. A. and P. Bühlmann (2009). On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics* 3, 1360–1392.
- Wainwright, M. (2009, May). Sharp thresholds for high-dimensional and noisy sparsity recovery using  $\ell_1$ -constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on* 55(5), 2183–2202.
- Zhao, P. and B. Yu (2006). On model selection consistency of lasso. *Journal of Machine Learning Research* 7, 2541–2563.
- Zhu, H., G. Leus, and G. B. Giannakis (2011). Sparsity-cognizant total least-squares for perturbed compressive sampling. *Signal Processing, IEEE Transaction on* 59(5).
- Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* 101(476), 1418–1429.